Defense Information Systems Agency

A Combat Support Agency

Tactical Edge Service:
Design Principles and Patterns

GIG EWSE Application and Service (EE212)

17 August 2011

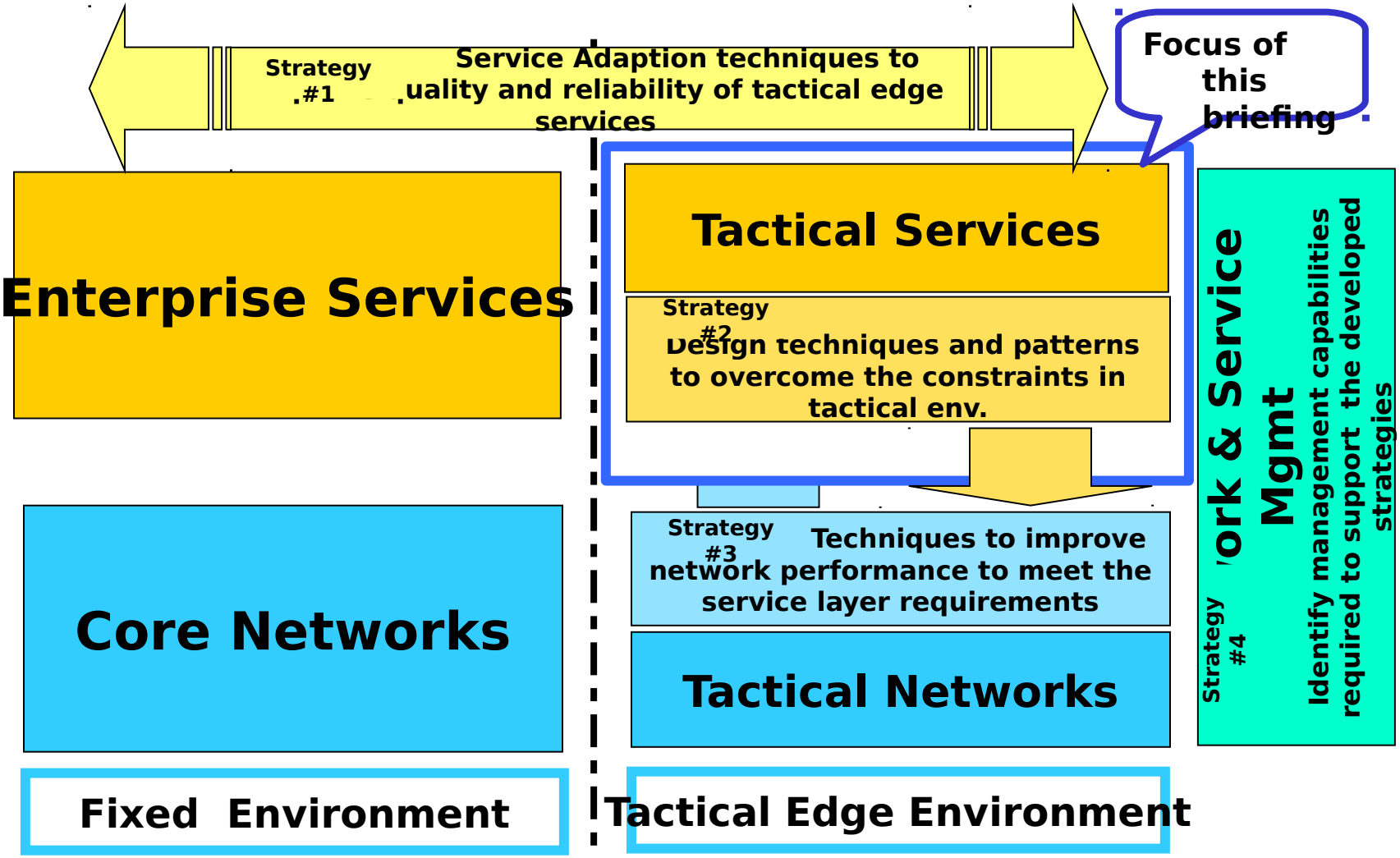
DISA
A Combat Support Agency

Outline

- Define Principles of Technical Approach
- Identify Design Techniques Overcoming Tactical Edge Constraints
- Show Example Technical Issues with Solutions
- Summarize Findings and Results
- Describe Way Ahead

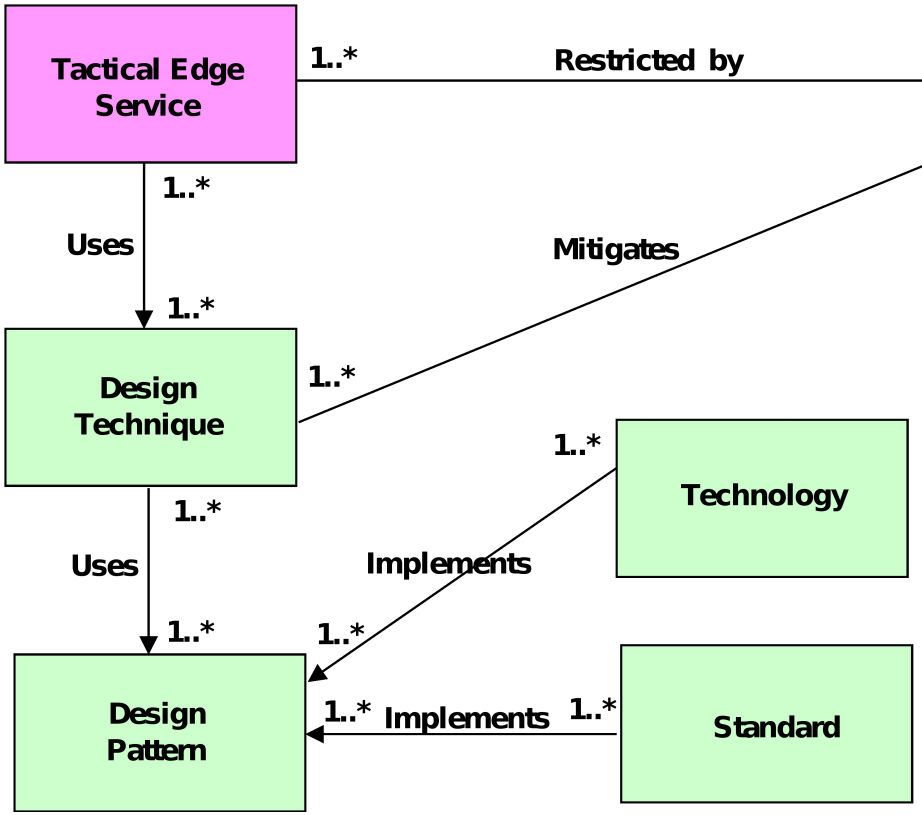
Technical Approach Framework

DISA
A Combat Support Agency


Strategy #1
Service Adaption techniques to
uality and reliability of tactical edge services
Focus of this briefing
Enterprise Services
Tactical Services
Strategy #2
Design techniques and patterns to overcome the constraints in tactical env.
ork & Service Mgmt
Strategy #4
Identify management capabilities required to support the developed strategies
Strategy #3
Techniques to improve network performance to meet the service layer requirements
Core Networks
Tactical Networks
Fixed Environment
Tactical Edge Environment

DISA
A Combat Support Agency

Principle of Technical Approach



- Consider tactical edge constraints as requirements for tactical services.
- Select a set of design techniques to mitigate and overcome tactical edge constraints for tactical services.
- Use design patterns to implement selected design techniques.
- Apply appropriate technologies and standards to implement design patterns.

DISA

A Combat Support Agency

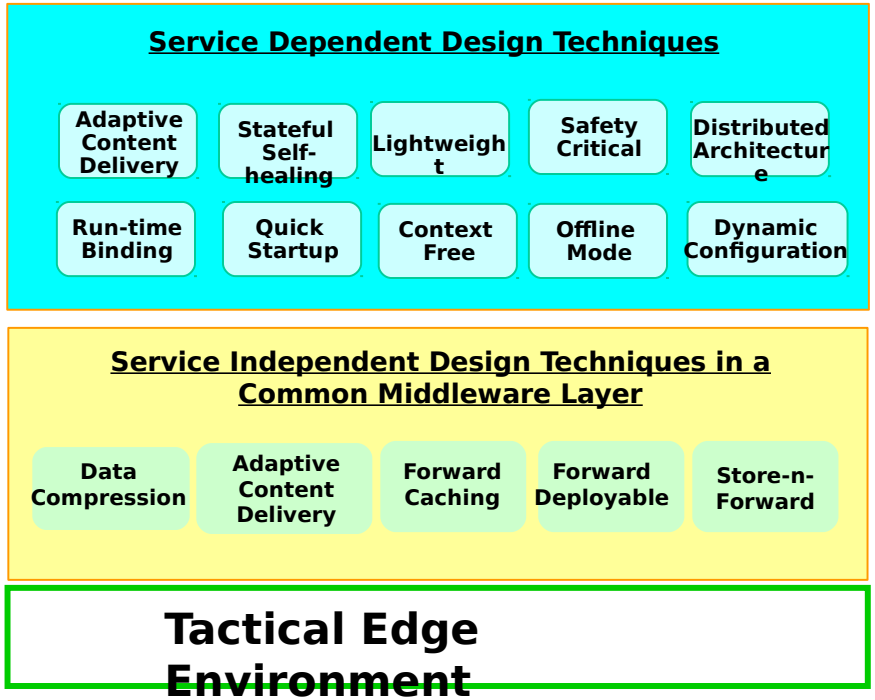
Summary of Design Techniques Overcoming Tactical Edge Constraints

Design Technique	Definition
Adaptive Content Delivery	Adaptive content delivery is the technique to deliver adapted content according to receiving device's capability profile, such as screen size, computing power, memory and storage capacity, OS, browser and network bandwidth. The less capable a receiving device, the lower the complexity of the content delivered to the device.
Context Free Service	Context free service makes a tactical edge service request independent of information from a prior request. Each request to the service must pass in all of the information the service needs to fulfill the request. The context free technique is mostly applicable to transactional services.
Distributed Architecture	A distributed architecture consists of multiple independent computing platforms hosting instances of the same tactical edge service deployed in different geographic locations that communicate to one another via IP network to support common capabilities in a decentralized manner.
Dynamic Configuration	Dynamic configuration enables a service and the computing platform hosting the service to automatically adapt to environmental changes, such as network connectivity changes due to the high mobility.
Data Compression	Data compression is the technique to encode information using fewer bits than the original representation of the information.
Forward Caching	Forward caching is the technique to preposition or store information in forward deployment positions in order to improve performance and reduce response time to access the information by services and/or GIG users at the tactical edge.
Forward Deployable	Forward deployable is the technique to package tactical edge services into a deployable package. The deployable package can be easily deployed to any compatible host platform in any forward deployment position.
Lightweight	Lightweight technique creates tactical edge services consuming minimal amount of computing, memory and storage resources.
Offline Mode	Offline mode allows a tactical edge service to continue to function even when the computing system hosting the tactical edge service lost network connectivity to the GIG. Once the computing system is reconnected, the tactical edge service would synchronize information with the GIG.
Quick Startup	Quick startup is the technique to make a tactical edge service ready for use nearly as soon as a GIG user starts the service (e.g., in a few seconds).
Run-time Binding	Run-time binding is the technique for a tactical edge service to bind with a called service only at run-time. As a result, it guarantees the binding with a called service instance available at run-time.
Stateful Self-healing	Stateful self-healing is the technique to make a tactical edge service automatically recover after an interruption, such as the temporary loss of network connection.
Safety Critical Service	Safety-critical service is the technique to develop a tactical edge service with limited capabilities, minimal resource requirements and high performance.
Store-n-forward	Store-n-forward is the technique to deliver messages of tactical edge services in a store-n-forward manner.

DISA

A Combat Support Agency

Service-dependent vs. Service- Independent Design Techniques



- Implement service dependent design techniques as part of a tactical service.

- Implement service independent design techniques in a middleware layer, which hides the complexity of mitigating tactical edge constraints from tactical services at higher layer.

- Adaptive Content Delivery can be implemented as part of a service or in a common middleware layer. The former allows greater optimization than the latter. However, the latter allows for greater reuse, potentially supporting a large number of tactical edge services.

DISA

A Combat Support Agency

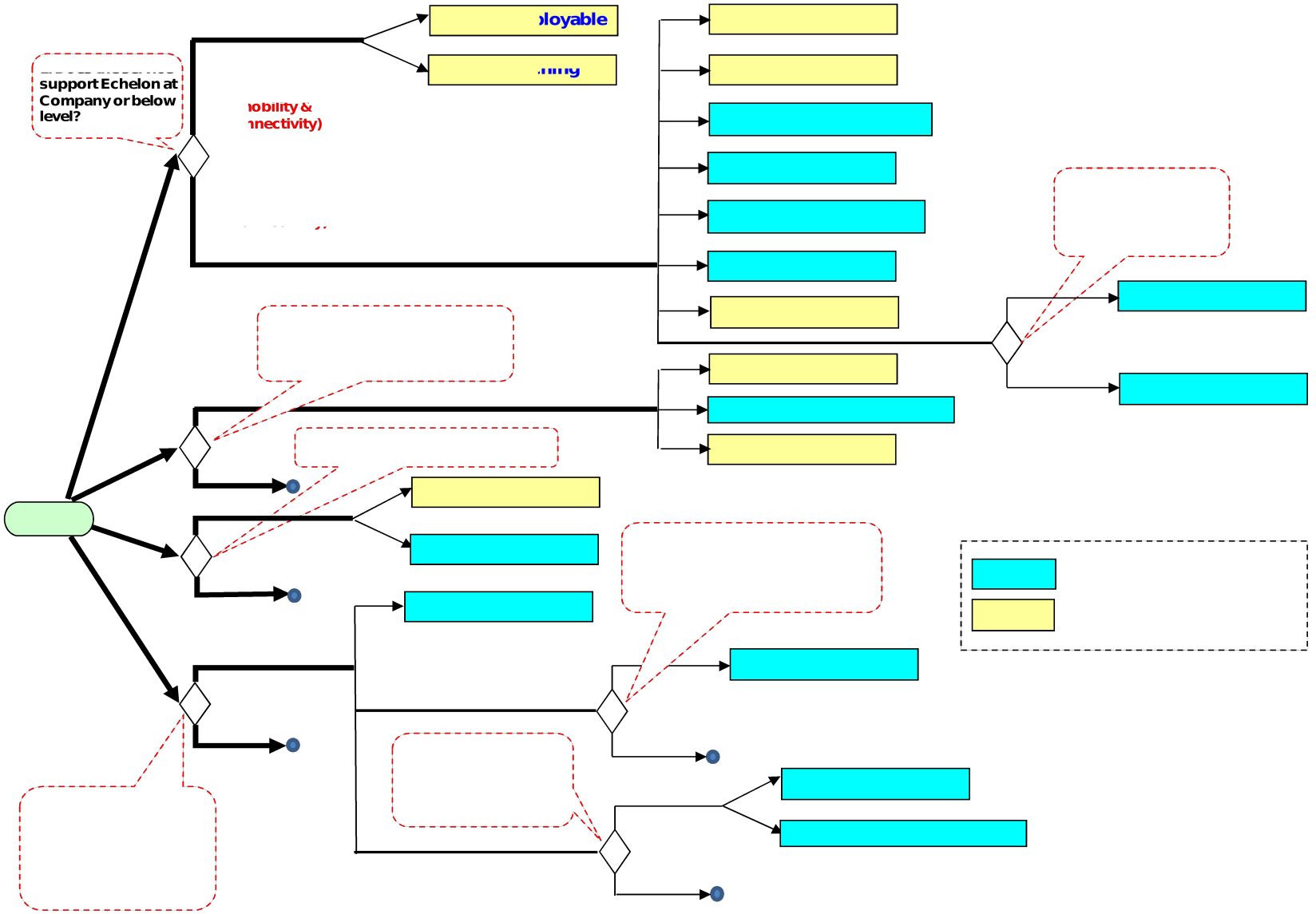
Mapping Design Techniques to Tactical Edge Constraints

Design Technique Overcoming Tactical Edge Constraints	gh Mobility	termittent Connectivity	sconnected	mited Bandwidth	ick Decision Time	ng Latency	mited Processing Power	inimal Storage	nall Weight	mited Power Supply	mplified Content	nall UI Size	of Constraints Addressed	ctical Fixed Center	ctical Mobile Center	bone Platform	smounted User
Forward Deployable	X	X	X	X	X	X							6	Useful	Important	Vital	Vital
Adaptive Content Delivery				X			X	X			X	X	5	Optional	Useful	Vital	Vital
Forward Caching	X	X	X	X		X							5	Optional	Important	Vital	Vital
Lightweight							X	X	X	X	X		5	Optional	Useful	Vital	Vital
Safety Critical					X		X	X	X	X			5	Useful	Useful	Vital	Vital
Distributed Architecture	X	X	X			X							4	Optional	Important	Vital	Vital
Run-Time Binding	X	X	X										3	Optional	Important	Vital	Vital
Stateful Self-Healing	X	X	X										3	Optional	Important	Vital	Vital
Context Free	X	X	X										3	Optional	Important	Vital	Vital
Store-n-Forward	X	X	X										3	Optional	Important	Vital	Vital
Dynamic Configuration	X	X											2	Optional	Important	Vital	Vital
Offline Mode	X		X										2	Optional	Important	Vital	Vital
Data Compression				X									1	Optional	Useful	Vital	Vital
Quick Startup					X								1	Useful	Useful	Vital	Vital
#of Techniques Addressing a Contraint	9	8	8	4	3	3	3	3	2	2	2	1					
Tactical Fixed Center	fixed	Connectivity: LAN >85%, WAN >99% Reliability: LAN >90%, WAN >90%		SAT: 8Mbps - nearly unlimited WAN: 16Mbps - nearly unlimited	mins - weeks	<250ms	nearly unlimited	nearly unlimited	100s lbs	grid, macro generator	complex	desktop - latptop		**Vital: the design technique mitigates one or more severe (red) constraints in the segment.**			
Tactical Mobile Center	occasional move	Connectivity: LAN >85%, WAN 85-99% Reliability: LAN >90%, WAN >75%		SAT: 128Kbps-4Mbps, WAN: 2-16Mbps(now),10-16Mbps(2013)	mins - days	LAN: <250ms, WAN: 250ms-1s	300-900 spec int	~10-50TB	100s lbs	generator - batteries	complex	desktop - latptop		**Important: the design technique mitigates two or more moderate (yellow) constraints in the**			
Mobile Platform	frequent move	Connectivity: LAN 25-84%, WAN 25-84% Reliability: LAN >90%, WAN <75%		SAT: 0.3Kbps-8Mbps (now), 128Kbps-8Mbps (2013) WAN: 2.6-486Kbps (now), 2.6Kbps-2Mbps (2013)	secs - mins	LAN: >250ms, WAN: 250ms-1s	<300 spec int	~500GB	10-100 lbs	generator - batteries	intermediate	laptop-tablet-handheld		**Useful: the design technique mitigates one moderate (yellow) constraint in the segment.**			
Dismounted User	frequent move	Connectivity: LAN 5-24%, WAN <5% Reliability: LAN >90%, WAN <75%		SAT: 0.3-60Kbps (now), 2.6-60Kbps (2013) WAN: 2.4-64Kbps (now), 2.4Kbps-0.6Mbps (2013)	secs - mins	LAN: >250ms, WAN: >1s	<30 spec int	~160GB	<10 lbs	batteries	simplified	laptop-tablet-handheld		**Optional: the design technique mitigates minimal or no constraints in the segment.**			

minimal to no constraints | moderate constraints | severe constraints

Decision Tree for Selecting Design Techniques for Tactical Service

A Combat Support Agency



DISA

A Combat Support Agency

Example of Selecting Design Techniques for UC at Tactical Ed

Question	Answer	Selected Design Technique
1. Does the service support Echelon at Company or below level?	Yes. Company, Platoon and below Therefore, the tactical edge environment will have severe constraints on high mobility, intermittent connectivity, and disconnected.	**Service dependent:** - Distributed architecture - Run-time binding - Dynamic configuration - Offline mode **Service independent:** - Forward deployable - Forward caching - Store-n-forward
1.1 Is this a transactional service using short-lived sessions?	No	**Service dependent:** - Stateful self-healing
2. Does the service require a high % of bandwidth available to the lowest supported Echelon?	Yes	**Service dependent:** - Adaptive content delivery **Service independent:** - Data compression
3. Does the service support quick decision time in secs?	N/A This is an always-on CES.	N/A
4. Does the service require a high % of processing power and storage resources on platforms in the lowest supported Echelon? For Platoon and below Echelon: Processing power constraint: 30 Spec Int Storage constraint: 160GB	No - Processing power requirement < 10 Spec Int -Storage requirement < 15 Mbps Therefore, it can be well supported by the available resources at the tactical	N/A

Summary of Design Techniques Selected for the Tactical Service

Service independent:	Service dependent:
- Forward deployable - Forward caching -Store-n-forward - Data compression	- Distributed architecture - Run-time binding - Dynamic configuration - Offline mode - Adaptive content delivery - Stateful self-healing

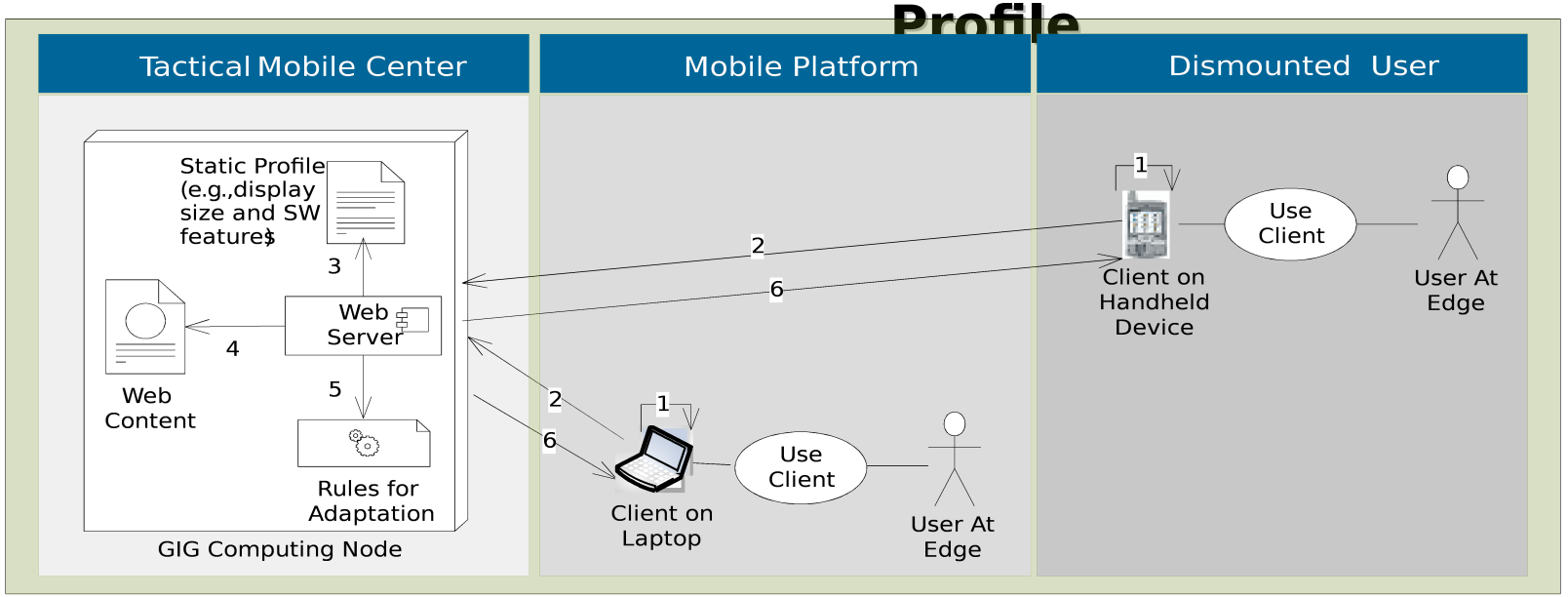
DISA
A Combat Support Agency

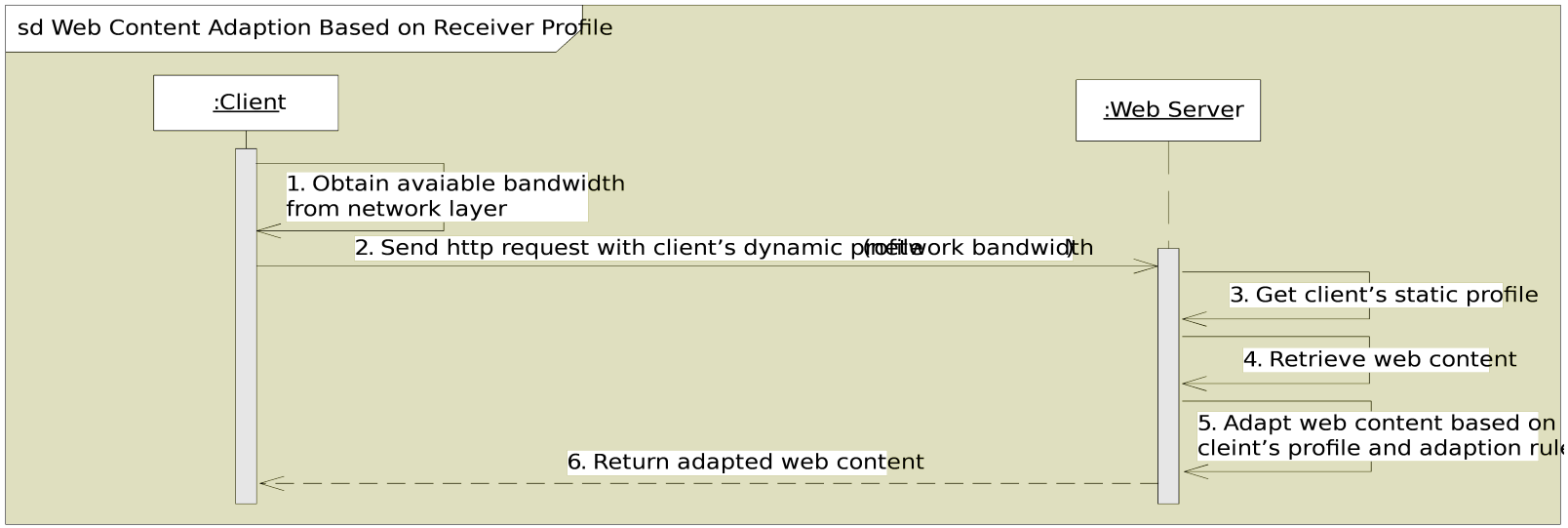
Display Problem On Non-Uniform Devices

- **Problem Description**
  - **Multimedia web pages fail to properly display on different types of handheld devices used by warfighters in the tactical edge environment.**
- **What Caused the Problem**
  - **Handheld devices can only support limited displaying capability due to its size, processing power and bandwidth constraints.**
- **Mitigation Techniques**
  - **Use Adaptive Content Delivery Technique.**
  - **Use Design Pattern: Web Content Adaption based on Receiver Profile.**

Design Pattern Example 1 – Web Content Adaption Based on Receiver Profile

A Combat Support Agency




sd Web Content Adaption Based on Receiver Profile
:Client
:Web Server
1. Obtain avaiable bandwidth from network layer
2. Send http request with client's dynamic profile (network bandwidth)
3. Get client's static profile
4. Retrieve web content
5. Adapt web content based on cleint's profile and adaption rule
6. Return adapted web content

DISA
A Combat Support Agency

Design Pattern Example 1

(Cont'd) – Web Content Adaption Based on Receiver Profile

- **Highlights of the Design Pattern**
  - **Profile capabilities of different tactical devices, such as display size, CPU type, storage capacity, software features, and network bandwidth.**
  - **Adapt web contents delivered to a tactical device based on its device profile.**
  - **This design pattern has been widely used in mobile industry.**
  - **This design pattern can be implemented using device profiling standards (such as Resource Description Framework from W3C) and web content adaption technologies (such as page layout conversion, pagination, multimedia conversion and resizing).**

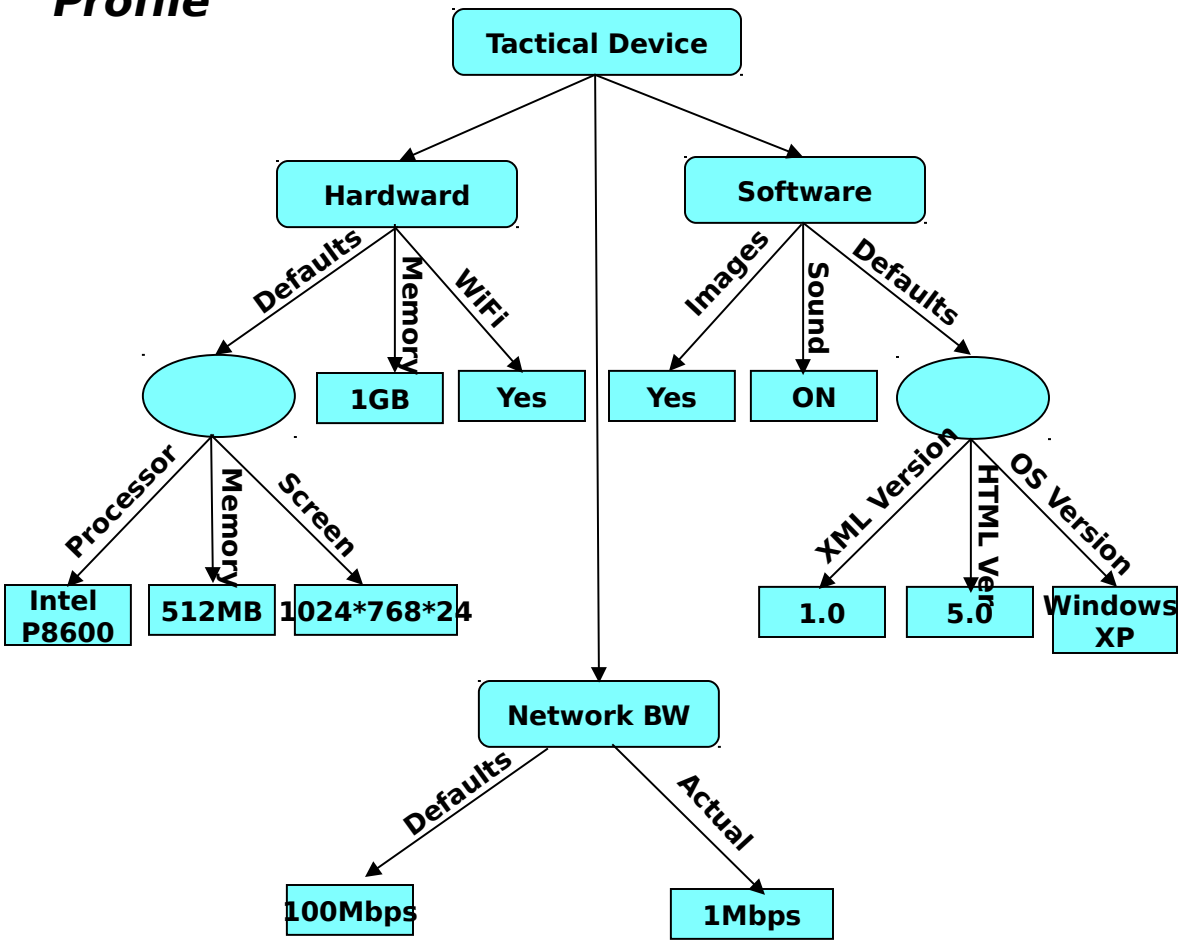
Design Pattern Example 1

(Cont'd) –

Web Content Adaption Based on Receiver Profile

A Combat Support Agency

Example Resource Description Framework Graph and XML File for Tactical Device Profile




```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#"
xmlns:prf="http://www.w3.org/TR/WD-profile-
vocabulary#">
 <rdf:Description about="HardwarePlatform">
   <prf:Defaults
    Vendor="Dell"
    Model="Latitude E6400"
    Type="Laptop"
    ScreenSize="1024x768x24"
    CPU="Intel P8600"
    Keyboard="Yes"
    Memory="512mB"
    Wifi="YES"
    Speaker="Yes" />
  <prf:Modifications
    Memory="1GB" />
 </rdf:Description>
 <rdf:Description about="SoftwarePlatform">
  <prf:Defaults
   OS="Windows XP"
   HTMLVersion="5.0"
   XMLVersion="1.0" />
  <prf:Modifications
   Sound="On"
   Images="Yes" />
 </rdf:Description>
 <rdf:Description about="Outlook2003">
   <prf:Defaults
   HTMLVersion="5.0" />
  </rdf:Description>
  <rdf:Description about="UserPreferences">
 <prf:Defaults
       Language="English"/>
 </rdf:Description>
</rdf:RDF>
<rdf:Description about="NetworkBW">
   <prf:Defaults
    Bandwidth="100Mbps"/>
 <prf:Modifications
    Bandwidth="1Mbps" />
 </rdf:Description>
```

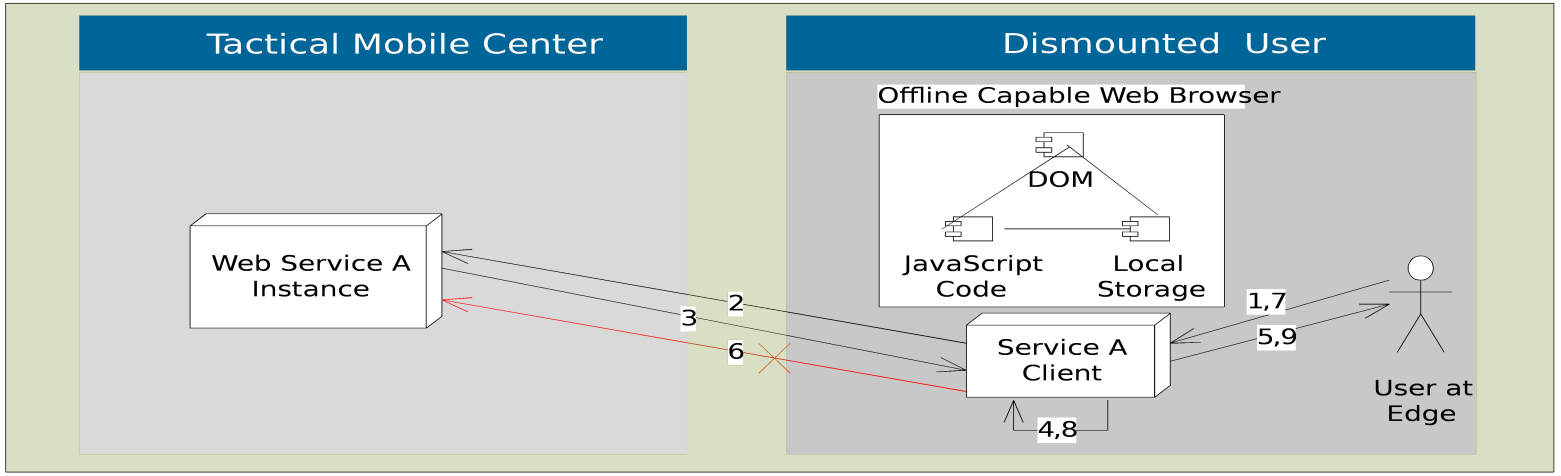
DISA
A Combat Support Agency

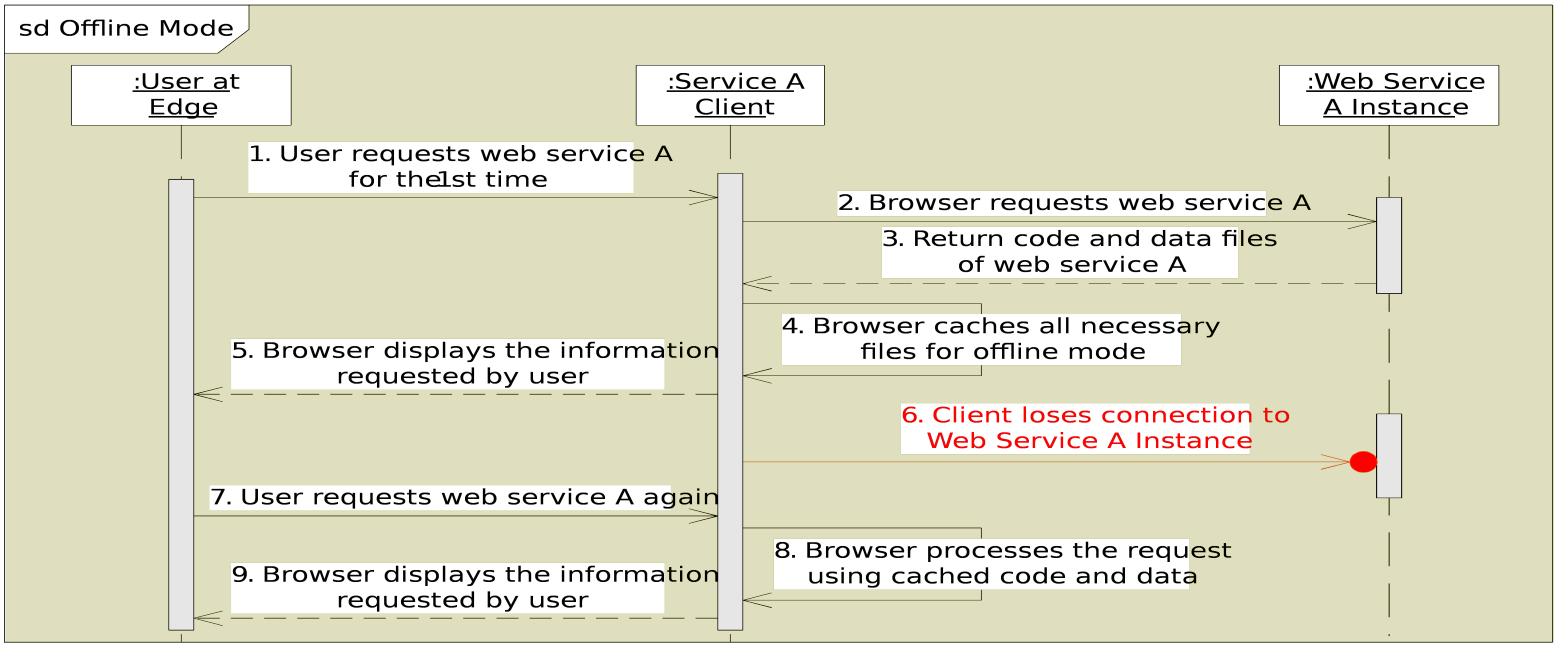
Unavailable Service Problem for Disconnected Warfighters

- **Problem Description**
  - **Application services stop working on disconnected devices used by warfighters in the tactical edge environment.**
- **What Caused the Problem**
  - **Clients lose connectivity to servers due to network constraints in the tactical edge environment.**
- **Mitigation Techniques**
  - **Use Offline Mode Technique.**
  - **Use Design Patterns: Offline Mode for Web Application & Database Synchronization.**

Design Pattern Example 2 – Offline Mode for Web Application

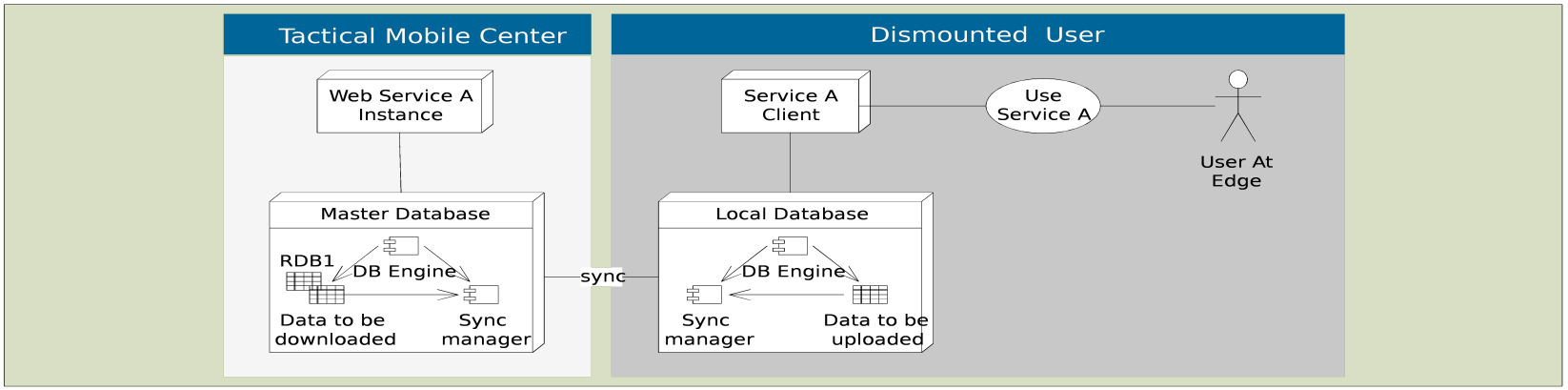
DISA
A Combat Support Agency

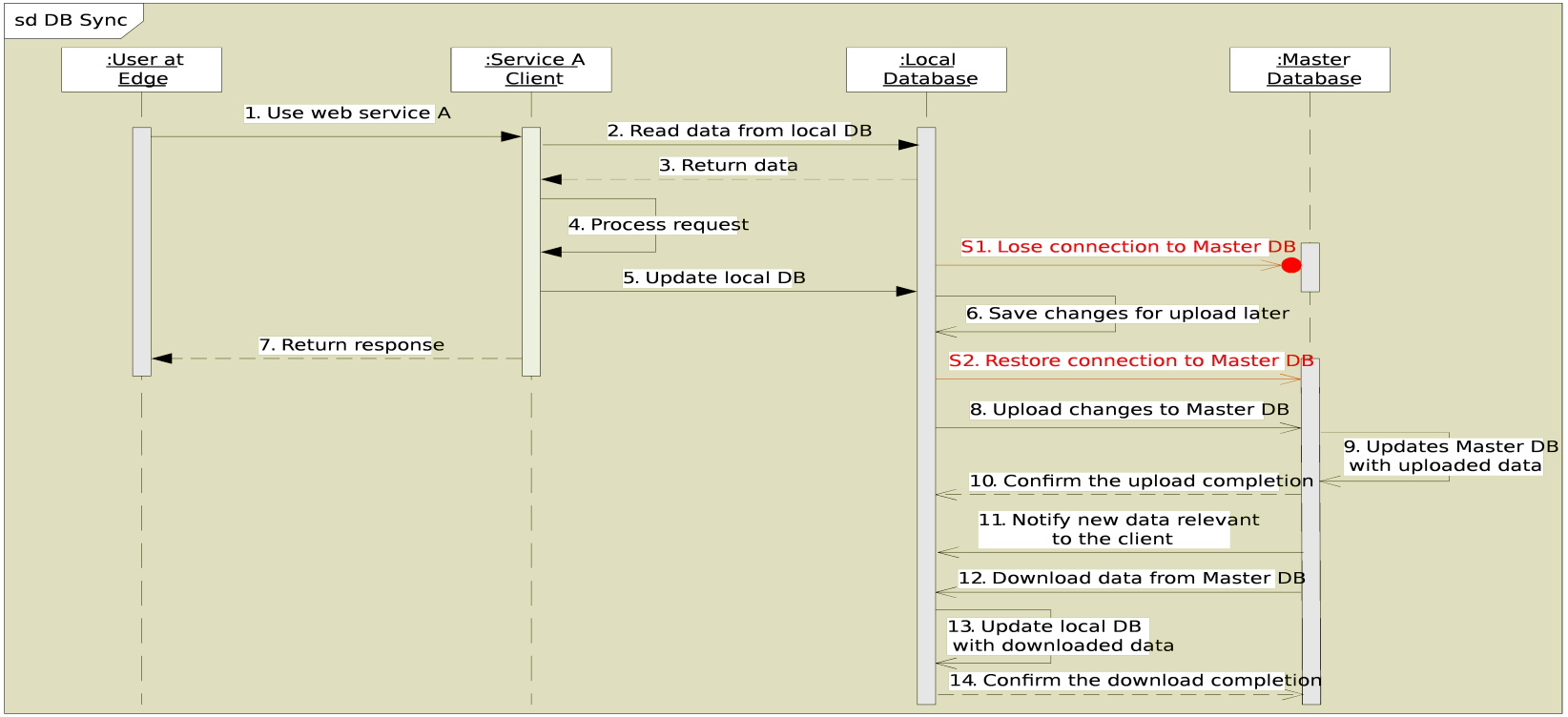



sd Offline Mode
:User at Edge
:Service A Client
:Web Service A Instance
1. User requests web service A for the1st time
2. Browser requests web service A
3. Return code and data files of web service A
4. Browser caches all necessary files for offline mode
5. Browser displays the information requested by user
6. Client loses connection to Web Service A Instance
7. User requests web service A again
8. Browser processes the request using cached code and data
9. Browser displays the information requested by user

Design Pattern Example 3 – Database Synchronization

A Combat Support Agency




sd DB Sync
:User at Edge
:Service A Client
:Local Database
:Master Database
1. Use web service A
2. Read data from local DB
3. Return data
4. Process request
S1. Lose connection to Master DB
5. Update local DB
6. Save changes for upload later
7. Return response
S2. Restore connection to Master DB
8. Upload changes to Master DB
9. Updates Master DB with uploaded data
10. Confirm the upload completion
11. Notify new data relevant to the client
12. Download data from Master DB
13. Update local DB with downloaded data
14. Confirm the download completion

DISA
A Combat Support Agency

Design Pattern Example 2 & 3 (Cont'd) – Offline Mode for Web Application & DB Synchronization

- **Highlights of the Design Patterns**
  - **Use client side scripting and local storage to build web applications which can continue to function even when the client loses connectivity to the server.**
  - **When losing connectivity to the server, the client would store new data entered by users locally. After reconnected with the server, the client would synchronize data with the server.**
  - **The design patterns have been widely used in mobile industry and have also been used in some DoD applications such as TIGR.**
  - **The design patterns can be implemented using the HTML5 standard from W3C, scripting technology (e.g., JavaScript), and lightweight database technology (such as SQLite).**

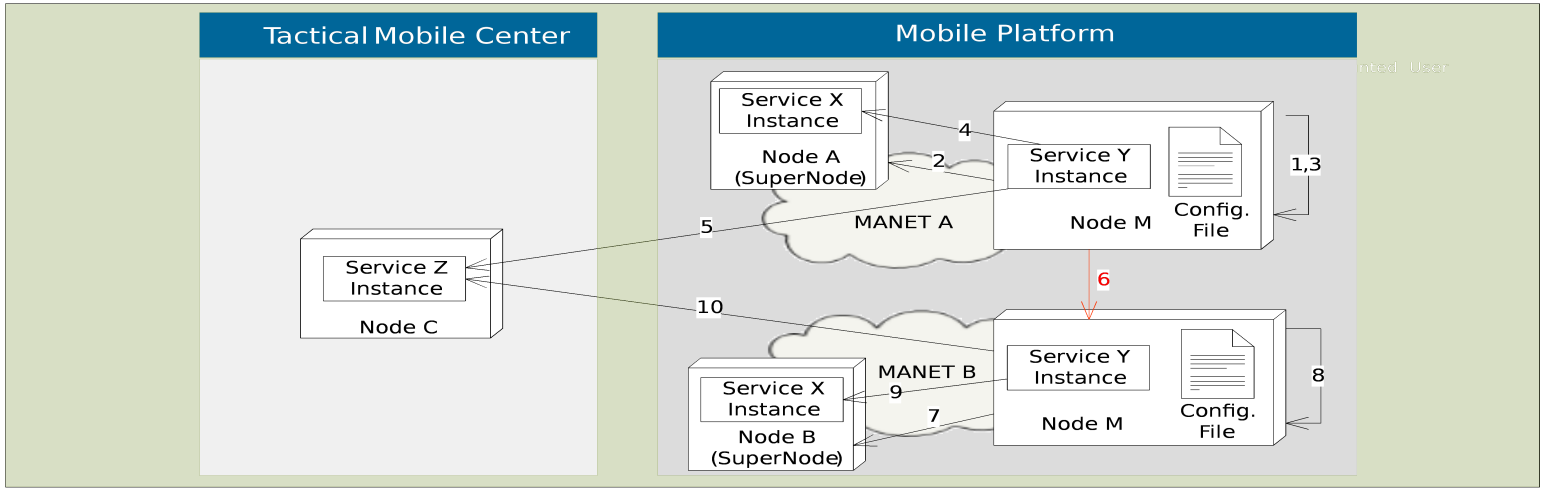
DISA
A Combat Support Agency

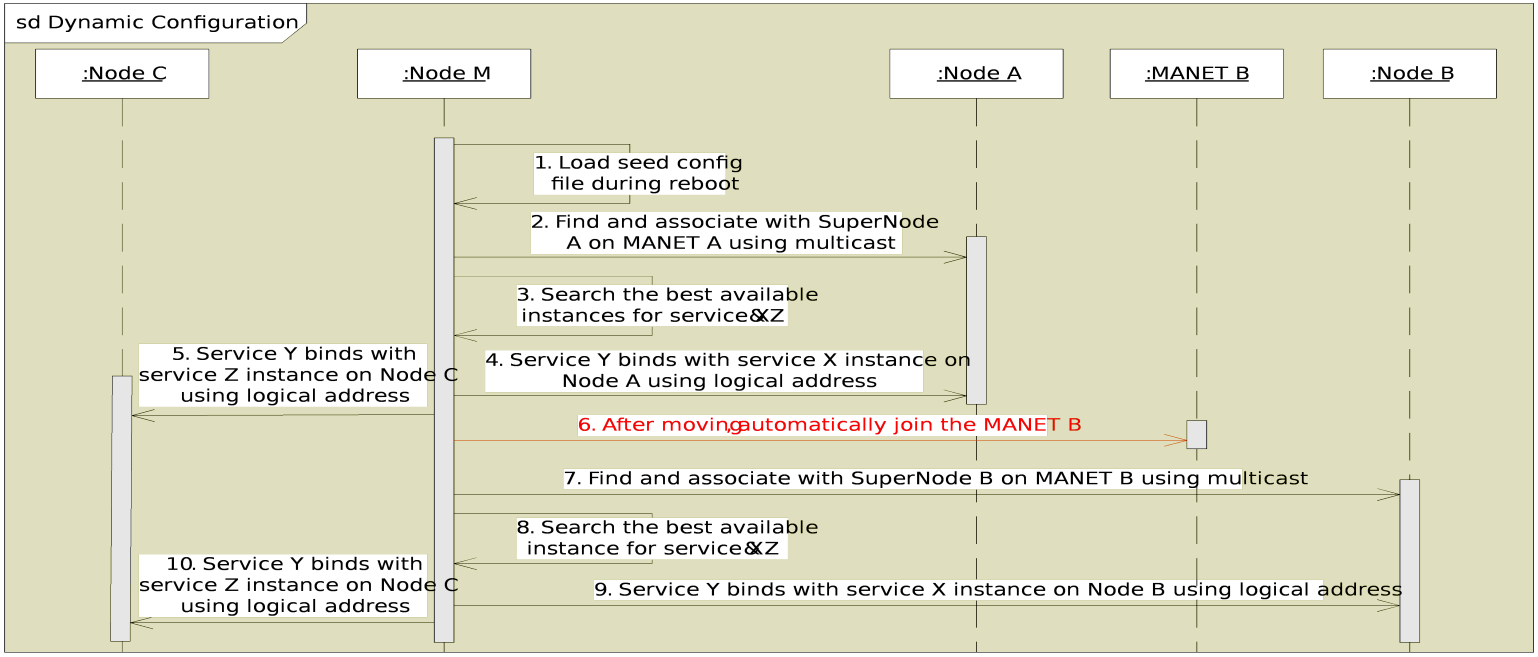
Service Disruption Problem on Moving Combat Vehicles

- **Problem Description**
  - **Services on combat vehicles often go into stalemate state when the vehicles move to new locations in the tactical edge environment.**
- **What Caused the Problem**
  - **When a combat vehicle moves to a new location, the vehicle may be disjointed from the distributed service network due to static configuration; services on the vehicle may also lose connection to the component service instances used in the old location.**
- **Mitigation Techniques**
  - **Use Dynamic Configuration and Run-time Binding Techniques.**
  - **Use Design Patterns: Dynamic Configuration & Run-time Binding.**

Design Pattern Example 4 – Dynamic Configuration

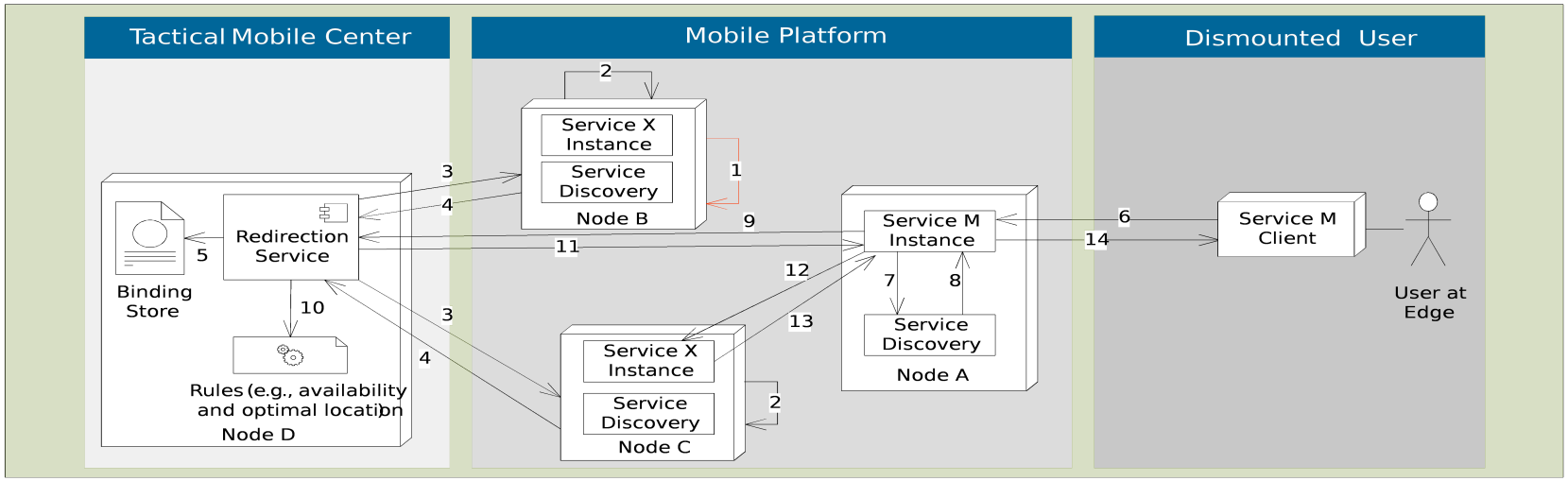
DISA
A Combat Support Agency

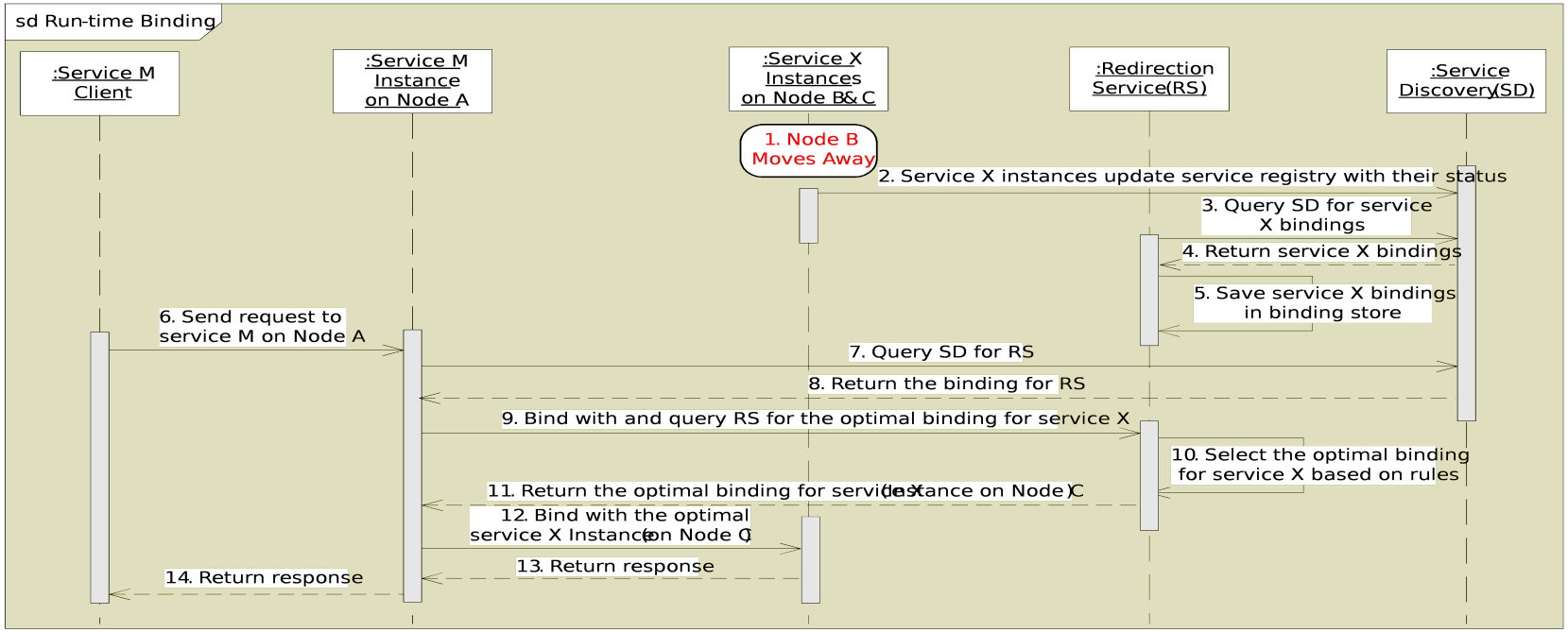



sd Dynamic Configuration
:Node C
:Node M
:Node A
:MANET B
:Node B
1. Load seed config file during reboot
2. Find and associate with SuperNode A on MANET A using multicast
3. Search the best available instances for service&Z
4. Service Y binds with service X instance on Node A using logical address
5. Service Y binds with service Z instance on Node C using logical address
6. After moving, automatically join the MANET B
7. Find and associate with SuperNode B on MANET B using multicast
8. Search the best available instance for service&Z
9. Service Y binds with service X instance on Node B using logical address
10. Service Y binds with service Z instance on Node C using logical address

Design Pattern Example 5 – Run-time Binding

DISA
A Combat Support Agency




sd Run-time Binding
:Service M Client
:Service M Instance on Node A
:Service X Instances on Node B& C
:Redirection Service(RS)
:Service Discovery(SD)
1. Node B Moves Away
2. Service X instances update service registry with their status
3. Query SD for service X bindings
4. Return service X bindings
5. Save service X bindings in binding store
6. Send request to service M on Node A
7. Query SD for RS
8. Return the binding for RS
9. Bind with and query RS for the optimal binding for service X
10. Select the optimal binding for service X based on rules
11. Return the optimal binding for service X (e.g. instance on Node C)
12. Bind with the optimal service X Instance on Node C
13. Return response
14. Return response

DISA
A Combat Support Agency

Design Pattern Example 4 & 5 (Cont'd) – Dynamic Configuration & Run-time Binding

- **Highlights of the Design Pattern**
  - **Dynamic configuration would be performed on both network layer (e.g., joining the nearby IP network) and application layer (e.g., joining the nearby service overlay network and dynamic service endpoint selection).**
  - **Run-time binding enables services to find and bind with optimal service endpoints according to pre-defined selection rules (such as optimal location and availability) at the run time.**
  - **Redirection service capability can be incorporated on multiple distributed service nodes.**
  - **The design patterns have been used by some DoD tactical edge PoRs, such as Army SOSCOE.**
  - **The design patterns can be implemented using SOA, DNS, IP multicast and other related technologies and standards.**

DISA
A Combat Support Agency

Summary

- Network, resource and operational constraints in tactical edge environment impose requirements for tactical service design and development.
- Design techniques can be used to mitigate and overcome tactical edge constraints.
- A design technique may be implemented using multiple design patterns/solutions based on different needs.
- A common set of design techniques and patterns for developing tactical services should be established at the DoD enterprise level to promote best practices and support interoperability among DoD PoRs.

DISA
A Combat Support Agency

Way Ahead

- **Establish tactical service design guidance in the DoD to drive the adoption of best practices and interoperability between implementations of different DoD tactical service PoRs.**
- **Improve and apply tactical service design framework by working with relevant DoD organizations, tactical service developer community and other DoD communities of interest (COIs).**
  - **Apply and validate framework to NCES capabilities.**
  - **Apply and validate the framework by other DoD COIs.**
  - **Continue improving the framework by collecting feedback and best practices from DoD COIs.**

DEFENSE INFORMATION SYSTEMS AGENCY